CUBIT Capability Proposal

Technical Area Geometry, Meshing, Infrastructure, GUI, Graphics, etc..	**Technical Lead** Cubit Developer in charge of technical area
Meshing	Matt Staten

MRD Description
Describe the capability in terms of how a user would see it.

Provide CAMAL API to basic sweeping transformation logic

SRS Description
What needs to be done by Cubit developers to implement this capability? Break the tasks into steps if applicable. (Steps should be on the order of 2 man-weeks or more)

Add a new interface to CAMAL:

```
CubitStatus transform_nodes
(
  int num_bnd_nodes,    /*IN – number of points on boundary loops*/
  double *src_bnd_nodes,   /*IN – location of src bnd points */
  double *trg_bnd_nodes,   /*IN – location of trg bnd points */
  int num_points_to_transform,   /*IN – number of points to transform*/
  double *points_to_transform,  /*IN – location of points to transform*/
  double *transformed_locations /*OUT – transformed locations of poins to transform*/
)
```

The node in points_to_transform are transformed with a transformation which approximates the step between src_bnd_nodes and trg_bnd_nodes.

Justification
Describe why this is important and what impact it will have if it is implemented. (or not implemented).

Unconstrained Plastering uses sweeplike logic to advance sheets inward. It would be nice if Unconstrained Plastering could leverage the transformation logic already in the sweeper. Currently CAMAL's interface to the sweeper requires the full connectivity of source quadrilaterals and returns full hexahedral connectivity. Unconstrained Plastering does not have quads to pass in, nor does it need the full hexahedral connectivity. Rather all it needs is the transformed location of the source interior nodes.

Resources Who will work on this	**Time estimate** How much time will it take in man-weeks	**Targeted Release** 10.2 (August 06), 10.3 (March 2007), 10.4 (August 2007), Future (beyond FY07)
Michael Stephenson	2 man-weeks	10.2

Submitted By:	**Date:**
Matt Staten	4/4/2006